

rechercher



actions - présentation

état: accepté

Présentation VisuAlea: Towards a Scientific Modelling Environment using Visual Programming

Présenté par [Christophe Pradal](#) (CIRAD, INRIA) dans [Scientific track 2010](#) le 2010/07/10 de 16:45 à 17:00 dans la salle Dussane

Résumé

Authors Christophe Pradal (1), Daniel Barbeau (2), Thomas Cokelaer (2),

Eric Moscardi (2)

Institutes: (1) CIRAD, (2) INRIA, Virtual Plants INRIA Team, UMR DAP, TA

A96/02, 34398 Montpellier Cedex 5, France

VisuAlea: Towards a Scientific Modelling Environment using Visual Programming

Data-flows and Visual Programming Environments (VPE) are used in many scientific domains such as data analysis and visualisation (LabView, VTK, VisTrails, Orange). One of the main interests of VPEs is the ability for users to create and modify dataflows using a graphical interface without any specific programming knowledge.

We developed OpenAlea (<http://openalea.gforge.inria.fr>) that is a python-based open-source software. It provides a visual programming environment, called VisuAlea , which is written in Python and PyQt. Within VisuAlea, users are able to dynamically combine existing and independent pieces of software

Pièces jointes



euroscipy2010_visualea_pradal.pdf

étiqueté par

15min_talk

biology

entités similaires

A Python Software Framework for Photonics Integrated

Circuits

A comparison of different approaches used to seek maintainability, performance and scalability in Python

scientific code

A Speech Recognition Toolkit based on Python

Teaching Scientific computing in Developing countries

Topographica: Python simulator for computational

neuroscience

Sponsors

Gold



Silver



ENTHOUGH

Bronze



Si vous souhaitez sponsoriser

cette conférence ou en être

partenaire veuillez nous

contacter

into a customizable dataflow. Although OpenAlea and VisuAlea are mainly used in the context of plant science, they can be efficiently used in other scientific domains.

A VPE that is implemented in Python language offers several interesting points for managing scientific workflows. First, users have access to the program state at run-time and thus can introspect, modify, document and debug their workflows interactively. Second, users can build graphically complex workflow by assembling both scientific python packages (e.g., numpy, scipy and matplotlib) with other software components written in different languages (C, C++, Fortran, Java or R). Finally, the visualisation of the workflow's structure ease the communication between different users. In VisuAlea, each component is provided with a graphical interface that is automatically generated. Moreover, dataflow model of computation allows to implement lazy evaluation, and automatic parallelisation.

VPEs have major drawbacks compared to textual programming (e.g., iteration structures are difficult to represent since dataflow is a directed acyclic graph). However, solutions may be found to cope with these issues. For instance, in VisuAlea, we have introduced functional loops like map to represent iteration structures.

Although several VPE have been implemented in Python (e.g. Orange, VisTrails, Viper), their visual environments are not reusable because functional and graphical parts are not separated. In OpenAlea, we have separated the dataflow graph, the evaluation algorithm and the GUI to allow

[voir tous]

other projects to reuse only the GUI part with their own data structure. The

OpenAlea.GraphEditor package implements the logic and the abstract

widgets to edit any types of graph such as dataflow without dependency on

other OpenAlea packages. Therefore, other projects can reuse OpenAlea.

GraphEditor with their own graph data structure via a pluggable system.

In this talk, we will briefly present the status of OpenAlea/VisuAlea, discuss

the advantages and drawbacks of a visual programming language, and

present some key functionalities of VisuAlea such as looping technics and

parallelism. Then, we will present the GraphEditor package with a brief

tutorial. Finally, we will present our methodology to manage the project (e.g.

deployment, documentation and quality assurance using continuous

integration and a compilation farm).

Présentation #1834 - dernière mise à jour 2010/06/06, créé le 2010/05/12 par Christophe Pradal

[Aide](#) | [À propos de ce site](#) | [Construit avec CubicWeb](#)